Assignment 2 64060

Nate Cvelbar

2023-10-01

```r
#install.packages("caret")
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```r
#install.packages("ISLR") # only install if needed
library(ISLR)
```

```r
#Assignment 2
#Nate Cvelbar
#BA-64060

#File taken online from course Assignment 2
#Loading the dataset
UB=read.csv('C:/Users/Owner/Documents/UniversalBank.csv')

#Mutate education and family

#install.packages("fastDummies") # only install if needed
library(fastDummies)
```

```
## Thank you for using fastDummies!
```

```
## To acknowledge our work, please cite the package:
```

```
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from
```

```r
UB<-dummy_cols(UB,select_columns=c("Family","Education"),remove_selected_columns = TRUE)


#Showing descriptive statistics
head(UB)
summary(UB)
```

```r
#Normalize Data
norm_mod<-preProcess(UB, method=c('range'))
UB_norm<-predict(norm_mod,UB)

#Dropping Zip Code and ID
UB_norm<-UB_norm[,-1]
UB_norm<-UB_norm[,-4]


#Move Personal Loan to front column
UB_norm<-UB_norm[,c(6, 1:5,7:17)]
head(UB_norm)

#Use 60% for training and rest for testing
Index_Train<-createDataPartition(UB_norm$Age, p=0.6, list=FALSE)
Train<-UB_norm[Index_Train,]
Test<-UB_norm[-Index_Train,]


TrainPre<-Train[,2:17]
TestPre<-Test[,2:17]

TrainLabels<-Train[,1]
TestLabels<-Test[,1]

library(class)


Predicticted_Test_labels<-knn(TrainPre, TestPre,cl=TrainLabels,k=1)

head(Predicticted_Test_labels)

#Set up predictor 1
Predictor1 = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer(84), CCA

Pre1<-knn(TrainPre,Predictor1,cl=TrainLabels,k=1,prob=TRUE)
attributes(Pre1)

#This customer would be classified as being predicted to accept the loan

set.seed(555)

Serach_grid<- expand.grid(k=c(1:10))
model<-train(Personal.Loan~Age+Experience+Income+CCAvg+Mortgage+Securities.Account+CD.Account+Online+Cr


## Warning in train.default(x, y, weights = w, ...): You are trying to do
## regression and your outcome only has two possible values Are you trying to do
## classification? If so, use a 2 level factor as your outcome column.

model

#k=3 is best fit
```

```r
#Confusion Matrix
#install.packages("gmodels")
library("gmodels")
CrossTable(x=TestLabels,y=Predicticted_Test_labels, prop.chisq=FALSE)


#Set up predictor 2
Predictor1 = data.frame(Age = as.integer(40), Experience = as.integer(10), Income = as.integer(84), CCA

Pre2<-knn(TrainPre,Predictor1,cl=TrainLabels,k=3,prob=TRUE)
attributes(Pre2)

#This customer would still be classified as being predicted to accept the loan
```